

The background is a complex collage of digital and technical imagery. On the left, a large, glowing orange network graph with interconnected nodes and lines is superimposed over a faint world map. The right side features a blue-toned interface with various data visualizations, including line graphs, bar charts, and control panels with buttons like 'EPID', 'LOGIC', 'EJECT', 'BIN', 'ON', and 'OFF'. Technical terms like 'VECTOR', 'COUNTED', and 'UNDEFINED' are also visible. The overall aesthetic is high-tech and data-driven.

# UTN Facultad Regional Resistencia

## Ingeniería en Sistemas de Información

### Algoritmos y Estructuras de Datos

**FUNCIONES, PROCEDIMIENTOS,  
PARÁMETROS Y ÁMBITOS DE VARIABLES**

# SUBACCIONES

# ACCION PRINCIPAL

**ACCION** [Nombre] (***PARÁMETROS***) **ES**

[**AMBIENTE**] → Definición de tipos, constantes y variables  
(**GLOBALES**)

[**SUB ACCIONES**] > Procedimientos y Funciones del algoritmo

**ALGORITMO** o [PROCESO]

[ACCIONES] → Algoritmo principal

**FIN ACCION**

# SUBACCIONES

**SUB-ACCION** [Nombre](*PARÁMETROS*): **[Tipo\_dato\_retorno]** **ES;**

**[AMBIENTE]** → Definición de tipos, constantes y variables

**(LOCALES)**

**ALGORITMO** o **[PROCESO]**

**[ACCIONES]** → Algoritmo principal de la sub-acción

**FIN SUB-ACCION**

# TIPO DE SUBACCIONES

Obtener el  
descuento  
que le  
corresponde  
al Cliente

**FUNCIONES**

Actualizar el  
interés  
acumulado

**PROCEDIMIENTOS**

# QUÉ SON LOS PROCEDIMIENTOS

- Un procedimiento es una sub-acción que ejecuta una tarea determinada.
- Son conjuntos de acciones agrupadas que se usan para evitar duplicación de código, conseguir soluciones más legibles y mantenibles.
- Permiten dividir un problema en sub-problemas logrando de esta manera escribir más fácilmente soluciones grandes y complejas.
- Pueden recibir valores de entrada (llamados parámetros).

# EJEMPLO DE PROCEDIMIENTOS

```
1  ACCION INTERESES ES
2
3  AMBIENTE
4      cuenta, dias: entero;
5      total: real;
6
7  [SUB-ACCIONES] <-- acá se definen
8
9  PROCESO
10     Pedir_Cuenta;
11     Procesar_intereses;
12     Imprimir;
13  FIN_ACCION.
```

# EJEMPLO DE PROCEDIMIENTOS

```
PROCEDIMIENTO Pedir_Cuenta ES
    ESCRIBIR('Datos de cuenta');
    LEER(cuenta, dias);
FIN_SUB;
```

```
PROCEDIMIENTO Procesar_intereses ES
    SI dias > 20 ENTONCES
        total := cuenta * 1,25;
    SINO
        total := cuenta * 1,15;
    FIN_SI;
FIN_SUB;
```

```
PROCEDIMIENTO Imprimir ES
    ESCRIBIR('Total de la cuenta = ', total);
FIN_SUB;
```



# QUÉ SON LAS FUNCIONES

- (ídem PROCEDIMIENTOS)

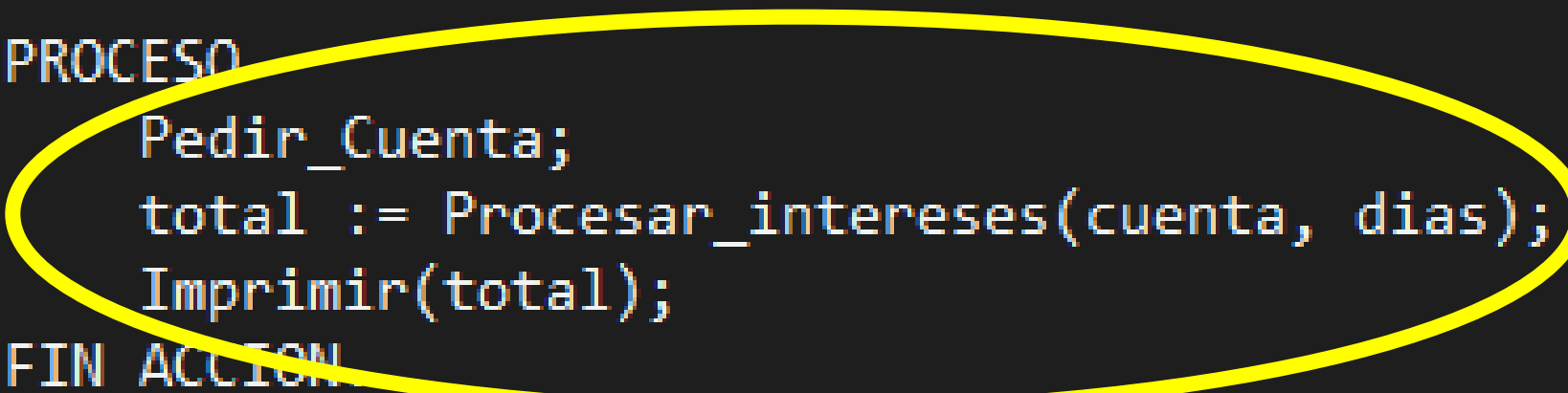
**Agregamos que:**

- Es una sub-acción que retorna un **valor de resultado**.



# EJEMPLO DE FUNCIÓN

```
1  ACCION INTERESES ES
2
3  AMBIENTE
4      cuenta, dias: entero;
5      total: real;
6
7  [SUB-ACCIONES] <-- acá se definen
8
9  PROCESO
10     Pedir_Cuenta;
11     total := Procesar_intereses(cuenta, dias);
12     Imprimir(total);
13  FIN_ACCION
```



# EJEMPLO DE FUNCIÓN

```
PROCEDIMIENTO Pedir_Cuenta ES
    ESCRIBIR('Datos de cuenta');
    LEER(cuenta, dias);
FIN_SUB;

FUNCION Procesar_intereses (p_cuenta, p_dias: entero): real ES
    SI p_dias > 20 ENTONCES
        Procesar_intereses := p_cuenta * 1,25;
    SINO
        Procesar_intereses := p_cuenta * 1,15;
    FIN_SI;
FIN_SUB;

PROCEDIMIENTO Imprimir (p_total: real) ES
    ESCRIBIR('Total de la cuenta = ', p_total);
FIN_SUB;
```

# TIPO DE PARÁMETROS

**POR VALOR**

**POR  
REFERENCIA**

# QUÉ SON LOS PARÁMETROS

A la función o procedimiento se le envía un valor que almacena en la variable correspondiente al parámetro (en el orden en que están definidos), la cual es **local**, de manera que su modificación no tiene efecto en el resto del algoritmo.

```
1  ACCION SUP_TRI ES
2
3  AMBIENTE
4      base, altura, sup: real;
5
6  FUNCION superficie_tri (p_base, p_altura: real): real ES
7      superficie_tri := p_base * p_altura / 2;
8  FIN_SUB;
9
10 PROCESO
11     LEER(base, altura)
12     sup := superficie_tri(base, altura);
13     ESCRIBIR(sup);
14 FIN_ACCION.
```

# ÁMBITO DE VARIABLES

Se utilizan  
en todo el  
algoritmo

**GLOBALES**

Se utilizan solo  
en la subacción  
que las define

**LOCALES**

# VARIABLES GLOBALES

Son variables definidas al comienzo del algoritmo (antes de cualquier sub-acción), que se pueden usar a lo largo de todo el algoritmo, es decir, dentro del algoritmo principal y en cada sub-acción definida en el algoritmo.

# VARIABLES LOCALES

Son variables definidas dentro de cada sub-acción, y que solo se pueden usar en la sub-acción, en la que son declaradas.

# LOCALES / GLOBALES

```
1  ACCION INTERESES ES
2
3  AMBIENTE
4      cuenta, dias: entero;
5      total: real;
6
7  [SUB-ACCIONES] <-- acá se definen
8
9  PROCESO
10     Pedir_Cuenta;
11     total := Procesar_intereses(cuenta, dias);
12     Imprimir(total);
13  FIN_ACCION.
```

Variables GLOBALES,  
(accesibles desde toda  
la solución)



# LOCALES / GLOBALES

```
PROCEDIMIENTO Pedir_Cuenta ES
    ESCRIBIR('Datos de cuenta');
    LEER(cuenta, dias);
FIN_SUB;

FUNCION Procesar_intereses (p_cuenta, p_dias: entero): real ES
    AMBIENTE
        porc: real;

    SI p_dias > 20 ENTONCES
        porc := 1,25;
    SINO
        porc := 1,15;
    FIN_SI;

    Procesar_intereses := p_cuenta * porc;
FIN_SUB;

PROCEDIMIENTO Imprimir (p_total: real) ES
    ESCRIBIR('Total de la cuenta = ', p_total);
FIN_SUB;
```

**Variables LOCALES,**  
**(accesibles solo desde**  
**la sub-acción)**